

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

**1.** (Previously presented) A method of creating network traffic replicating activities of a large number of users comprising:

receiving a test script including a plurality of commands specifying extended operations of an operating system

invoking a script interpreter

launching an application thread to execute the test script, wherein the application thread is a light weight thread capable of invoking the extended operations, the application thread requiring a smaller amount of memory, less processor power, and less communication with the operating system when compared to traditional threads

invoking a protocol engine for each of the commands in the test script such that each protocol engine has an associated command,

each protocol engine executing its associated command.

**2.** (Original) The method of claim 1 wherein the commands in the test script simulate actions taken by a network user.

**3.** (Cancelled)

**4.** (Previously presented) The method of claim 1 wherein the extended operations include “fetch,” “verify,” “fetch and verify,” “fetch and ignore,” “monitor,” and “count.”

5. (Original) The method of claim 1 wherein the test script causes network traffic to be produced.
6. (Original) The method of claim 1 wherein each protocol engine executing its associated command comprises:

checking whether a maximum number of protocol engines has been exceeded

performing the executing when the maximum number of protocol engines has not been exceeded.
7. (Previously presented) The method of claim 6 wherein the checking further comprises:

waiting a system defined amount of time until attempting to execute again.
8. (Original) The method of claim 6 wherein the checking further comprises:

sleeping until system resources sufficient for the executing of the protocol engine are available until attempting to execute again.
9. (Original) The method of claim 1 wherein the network traffic is comprised of a plurality of data units adhering to a plurality of communications protocols.
10. (Original) The method of claim 9 wherein the plurality of communication protocols includes at least one of Ethernet, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Internet Protocol (IP), File Transfer Protocol (FTP), or Hypertext Transfer Protocol (HTTP).

**11.** (Currently amended) A non-transitory computer machine readable medium having instructions stored thereon which when executed cause a processor to perform operations comprising:

receiving a test script including a plurality of commands specifying extended operations of an operating system

invoking a script interpreter

launching an application thread to execute the test script, wherein the application thread is a light weight thread capable of invoking the extended operations, the application thread requiring a smaller amount of memory, less processor power, and less communication with the operating system when compared to traditional threads

invoking a protocol engine for each of the commands in the test script such that each protocol engine has an associated command,

each protocol engine executing its associated command.

**12.** (Currently amended) The non-transitory computer machine readable medium of claim 11 wherein the commands in the test script simulate actions taken by a network user.

**13.** (Cancelled)

**14.** (Currently amended) The non-transitory computer machine readable medium of claim 11 wherein the extended operations include “fetch,” “verify,” “fetch and verify,” “fetch and ignore,” “monitor,” and “count.”

**15.** (Currently amended) The non-transitory computer machine readable medium of claim 11 wherein the test script causes network traffic to be produced.

**16.** (Currently amended) The non-transitory computer machine readable medium of claim 11 wherein each protocol engine executing its associated command comprises:

    checking whether a maximum number of protocol engines has been exceeded  
    performing the executing when the maximum number of protocol engines has not been exceeded.

**17.** (Currently amended) The non-transitory computer machine readable medium of claim 16 wherein the checking further comprises:

    waiting a system defined amount of time before attempting to execute again.

**18.** (Currently amended) The non-transitory computer machine readable medium of claim 11 coupled with a network testing system.

**19.** (Currently amended) The non-transitory computer machine readable medium of claim 18 wherein the network testing system is coupled to a production network.

**20.** (Currently amended) The non-transitory computer machine readable medium of claim 19 wherein the network testing system is coupled to a test network.

**21.** (Previously presented) A system to create network traffic simulating activities of a large number of users, the system comprising:

a plurality of script interpreter units in user space, each script interpreter unit to interpret a script including a plurality of commands, the commands specifying extended operations of an operating system

an application thread in user space for each script interpreter unit, wherein each application thread is a light weight thread capable of invoking the extended operations, each application thread requiring a smaller amount of memory, less processor power, and less communication with the operating system when compared to traditional threads

a plurality of protocol engines in user space for each application thread, each protocol engine to execute a command included in one of the scripts

an operating system in operating system space

an I/O multiplexor between the operating system in operating system space and the protocol engines in user space, the I/O multiplexor operating in user space and operating system space and configured so that the protocol engines do not have to wait for or block on the operating system, the I/O multiplexor configured to

receive calls to the operating system from the protocol engines and direct the calls to the operating system

receive responses from the calls placed with the operating system from the operating system and pass the responses an appropriate protocol engine.

**22.** (Original) The system of claim 21 wherein the system supports a plurality of communications protocols.

**23.** (Original) The system of claim 22 wherein the plurality of communications protocols includes at least Ethernet, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Internet Protocol (IP), and Hypertext Transfer Protocol (HTTP).

**24.** (Cancelled)

**25.** (Previously presented) The system of claim 21 wherein the extended operations include “fetch and verify” and “fetch and ignore.”

**26.** (Previously presented) A system to create network traffic simulating activities of a large number of users, the system comprising:

    a plurality of script interpreter units in user space, each script interpreter unit to interpret a script including a plurality of commands, the commands specifying extended operations of an operating system

    an application thread in user space for each script interpreter unit, wherein each application thread is a light weight thread capable of invoking the extended operations, each application thread requiring a smaller amount of memory, less processor power, and less communication with the operating system when compared to traditional threads

    a plurality of protocol engines in operating system space for each application thread, each protocol engine to execute a command included in one of the scripts

    an operating system in operating system space

an I/O multiplexor between the operating system and the protocol engines, the I/O multiplexor configured so that the protocol engines do not have to wait for or block on the operating system, the I/O multiplexor configured to

receive calls to the operating system from the protocol engines and direct the calls to the operating system

receive responses from the calls placed with the operating system from the operating system and pass the responses an appropriate protocol engine.

**27.** (Original) The system of claim 26 wherein the system supports a plurality of communications protocols.

**28.** (Original) The system of claim 27 wherein the plurality of communications protocols include one or more of Ethernet, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Internet Protocol (IP), and Hypertext Transfer Protocol (HTTP).

**29.** (Cancelled)

**30.** (Previously presented) The system of claim 26 wherein the extended operations include “fetch and verify” and “fetch and ignore.”

**31.** (Previously presented) A system to create network traffic simulating activities of a large number of users, the system comprising:

a plurality of script interpreter units in user space, each script interpreter unit to interpret a script including a plurality of commands, the commands specifying extended operations of an operating system

an application thread in operating system space for each script interpreter unit, wherein each application thread is a light weight thread capable of invoking the extended operations, each application thread requiring a smaller amount of memory, less processor power, and less communication with the operating system when compared to traditional threads

a plurality of protocol engines in operating system space for each application thread, each protocol engine to execute a command included in one of the scripts

an operating system in operating system space

an I/O multiplexor between the operating system and the protocol engines, the I/O multiplexor configured so that the protocol engines do not have to wait for or block on the operating system, the I/O multiplexor configured to

receive calls to the operating system from the protocol engines and direct the calls to the operating system

receive responses from the calls placed with the operating system from the operating system and pass the responses an appropriate protocol engine.

**32.** (Original) The system of claim 31 wherein the system supports a plurality of communications protocols.

**33.** (Original) The system of claim 32 wherein the plurality of communications protocols includes at least one of the following: Ethernet, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Internet Protocol (IP), and Hypertext Transfer Protocol (HTTP).

**34.** (Cancelled)

**35.** (Previously presented) The system of claim 31 wherein the extended operations include “fetch and verify” and “fetch and ignore.”